Repository - Bug #6541

repostory: unlink objects failure

12/08/2013 11:31 - Anonymous

Status:	Bug resolved	**Start date:**	12/08/2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	LCMS 4 Beta	**Spent time:**	0.00 hour
Complexity:	Normal		

Description

when unlinking objects in the repository: "Your call to the method escape_column_name in repository\ColumnNameDataManager could not be resolved."

History

#1 - 26/08/2013 16:38 - Anonymous

- Target version set to LCMS 4 Beta

#2 - 30/08/2013 15:32 - Anonymous

- Status changed from New to Needs testing

the method ***escape_column_name*** is called in many different ways:

```
\repository\DataManager :: escape_column_name

\repository\DataManager :: get_instance() -> escape_column_name
$this->escape_column_name
```

The first one is erroneous, and appears in files

```
application/elude/php/lib/data_manager/mdb2.class.php
application/gutenberg/php/lib/data_manager/mdb2.class.php
application/internship_organizer/php/lib/data_manager/database_internship_organizer_data_manager.class.php
application/internship_organizer/php/lib/data_manager/mdb2.class.php
application/package/php/lib/data_manager/mdb2.class.php
application/photo_gallery/php/lib/data_manager/mdb2.class.php
application/phrases/php/lib/data_manager/mdb2.class.php
application/profiler/php/lib/data_manager/mdb2.class.php
application/wiki/php/lib/data_manager/mdb2.class.php
```

The only places where a method ***escape_column_name*** is defined are in the Dcotrine and Mdb2 databse classes. I believe the correct use is therefor

```
$this->escape_column_name
```

in the derived classes.

#3 - 30/08/2013 15:35 - Anonymous

- Status changed from Needs testing to Needs more info

#4 - 30/08/2013 15:39 - Anonymous

- Project changed from Courses to Repository

#5 - 02/09/2013 10:56 - Anonymous

I failed to see the method escape_column_name is static. The most correct use is: ***Mdb2Database*** :: *escape_column_name*, or ***DoctrineDatabase*** :: *escape_column_name*.

However, given PHP's non-strict evaluation, calling a static method non-statically is harmless.

#6 - 02/09/2013 11:31 - Anonymous

- Status changed from Needs more info to Bug resolved

fixed

removed all incorrect ***repository*****DataManager*** :: *escape_column_name* calls and replaced with ***Mdb2Database*** :: *escape_column_name*

#7 - 02/09/2013 11:38 - Sven Vanpoucke

- Status changed from Bug resolved to Assigned

- Assignee set to Anonymous

Never replace a call to the general DataManager to one of it's implementations. You should always call DataManager :: get_instance() if you need to call a function of your implementation (depending on the currently selected implementation in the config).

Please revert this last change to \repository\DataManager :: get_instance()->escape_column_name

#8 - 02/09/2013 11:44 - Anonymous

What should I do with calls like $this->escape_column_name()? They are all over the place

#9 - 02/09/2013 11:50 - Anonymous

And why is *escape_column_name* a static method? One needs to call it through a singleton, while the datamanagers extend directly from the Mdb2 or Doctrine base classes? This seems like a useless detour, adding complexity but no functionality.

The codebase contains a lot of calls to ***DoctrineDatabase*** :: *escape_column_name*, which I took as a reference for this bugfix. Should I replace all of them too?

#10 - 02/09/2013 12:04 - Sven Vanpoucke

There is a difference from what source you are working:

If you are working from an MDB2 or Doctrine implementation you should call parent :: escape_column_name since it's an extension
If you are working from the general DataManager you should call $this->get-instance()->escape_column_name
If you are working from an external source like a component you should call DataManager :: get_instance()->escape_column_name

But never reference to the MDB2Database or DoctrineDatabase directly.

#11 - 02/09/2013 13:23 - Anonymous

it remains inconsistent: why is is called statically in the first case and non-statically in the other cases?

#12 - 02/09/2013 13:36 - Sven Vanpoucke

Because the DataManager class is basically a delegation class with some additional functionality (like caching), which redirects most of it's logic to the selected implementation through the Singleton pattern. This is a static class and therefor the delegation function for the escape_column_name is also static. It is there so you can call this function without the need to execute the get_instance(). Basically it's just a facade for DataManager :: get_instance()->escape_column_name

#13 - 02/09/2013 13:41 - Anonymous

I guess that classes like **common/libraries/php/shared/storage/implementation/doctrine/condition/equality_condition.class.php** can still use ***DoctrineDatabase*** :: *escape_column_name*, as they are by definition bound to Doctrine? (analogous for Mdb2).

I am currently fixing 1166 instances of the escape_column_name problem. It seems unreasonable that all of them are wrong, however they do not comply to the rules you explained above.

#14 - 02/09/2013 13:47 - Sven Vanpoucke

They are core classes which are already inside the doctrine implementation so they obviously belong to the doctrine implementation and therefore these classes are the only ones that do not need to comply to the rules above. But since this does not belong to the broken functionality, you should not bother looking into these classes.

#15 - 02/09/2013 14:13 - Anonymous

I am still not satisfied: the only reason it works with this scheme is because PHP allows to call static functions in a non-static way. I fail to see the point of mixing static and non-static. Anyway, I am not in a position to question the design, I'm just pointing out the inconsistency.

#16 - 02/09/2013 14:39 - Anonymous

- Status changed from Assigned to Needs more info

#17 - 03/09/2013 12:02 - Anonymous

- Status changed from Needs more info to Bug resolved

Files

Screenshot from 2013-08-12 11_29_42.png	68.7 KB	12/08/2013	Anonymous